

Pedro Celis

**SQL Server 9.0 Storage Engine (Yukon) Overview
For Non-Database Researchers**

Presentation Goals

- Describe where we've come from
- Overview of Yukon scope
- Some technical details on Yukon
- Solicit feedback
- Initiate ongoing dialog

Where we've come from

SQL 7.0

- We mostly knew what we were going to build.
- Needed to make sure what we built worked:
 - New QP, Parallelism, new storage engine

Where we've come from

SQL 7.0

- We mostly knew what we were going to build.
- Needed to make sure what we built worked:
 - New QP, Parallelism, new storage engine

SQL 2000

- Clean up and several new features
- (We thought we'd need a new release quickly - we were wrong...)

Where we've come from

● SQL 7.0

- We mostly knew what we were going to build.
- Needed to make sure what we built worked:
 - New QP, Parallelism, new storage engine

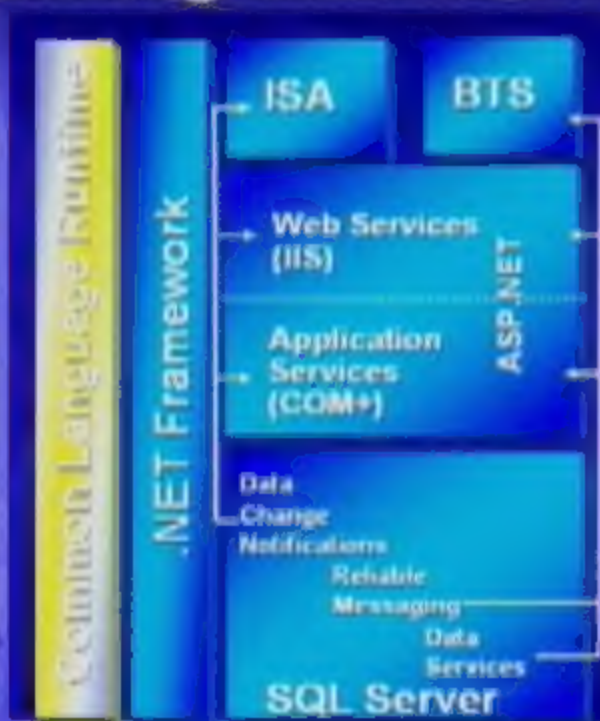
● SQL 2000

- Clean up and several new features
- (We thought we'd need a new release quickly - we were wrong...)

● Yukon

- Innovation!
- Continued evolution of architecture established in 7.0

Winning with .NET Servers

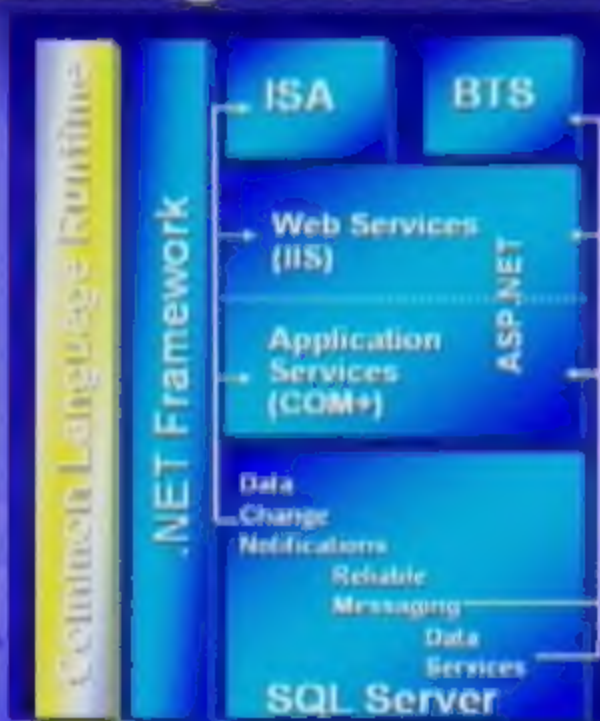


Windows

"Better Together"

- **CLR**
 - Language Freedom
- **XML/SOAP**
 - Open protocol
- **Caching**
 - All Levels
- **Diagnostics**
 - All Levels
- **Security**
 - Unifying concepts
- **Management**
 - SLA focus

Winning with .NET Servers

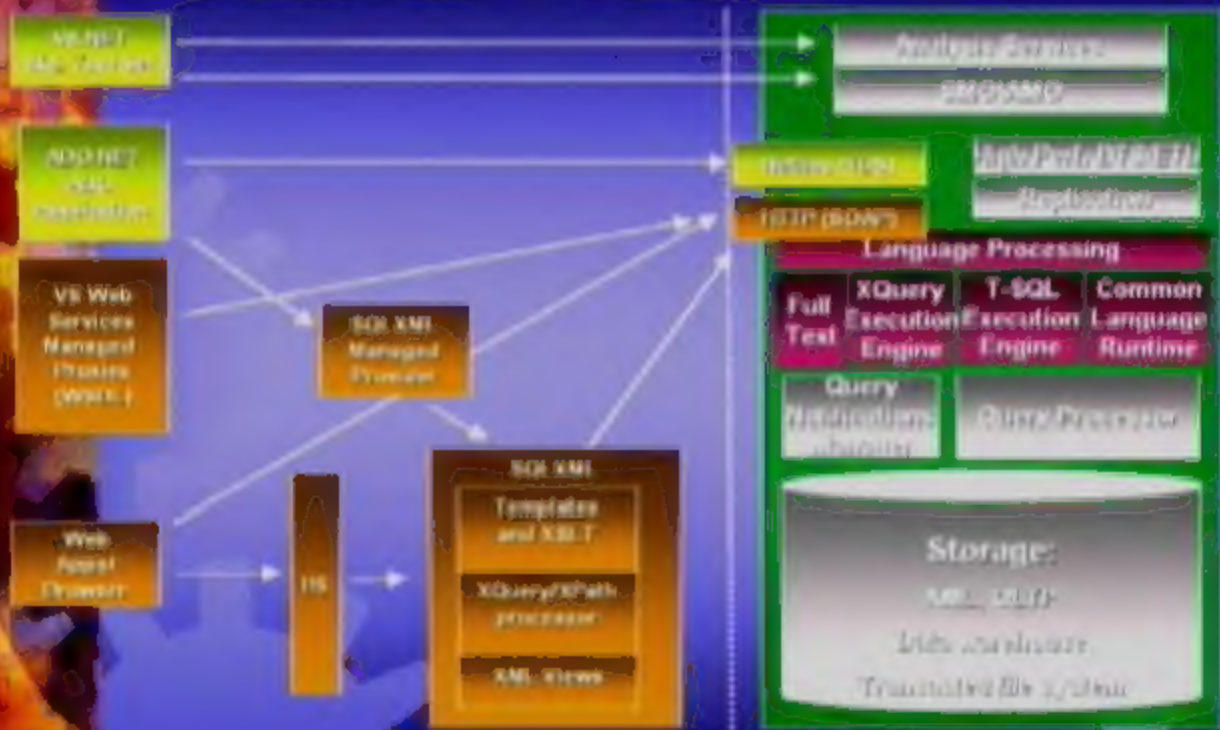


Windows

"Better Together"

- **CLR**
 - Language Freedom
- **XML/SOAP**
 - Open protocol
- **Caching**
 - All Levels
- **Diagnostics**
 - All Levels
- **Security**
 - Unifying concepts
- **Management**
 - SLA focus

SQL Server "Yukon"



Yukon Server Investment

- Scalability - Big scale up investment
 - Partitions
 - Online index build
 - SMP enhancements
- Availability - Many new features
 - Online partial database restore
 - Real-time log shipping

Yukon Server Investment

Project Summary

- 11/15/2000 - 11/16/2000
- 11/17/2000 - 11/18/2000
- 11/19/2000 - 11/20/2000
- 11/21/2000 - 11/22/2000
- 11/23/2000 - 11/24/2000
- 11/25/2000 - 11/26/2000

Caveat

- **Stressors** are factors that cause stress
- **Stressors** can be physical, chemical, or psychological
- **Stressors** can be acute or chronic
- **Stressors** can be internal or external
- **Stressors** can be positive or negative
- **Stressors** can be beneficial or harmful
- **Stressors** can be a source of growth or a source of damage
- **Stressors** can be a challenge or a threat
- **Stressors** can be a demand or a pressure
- **Stressors** can be a change or a transition
- **Stressors** can be a loss or a gain
- **Stressors** can be a conflict or a choice
- **Stressors** can be a risk or a reward
- **Stressors** can be a goal or a dream
- **Stressors** can be a hope or a despair
- **Stressors** can be a faith or a doubt
- **Stressors** can be a love or a hate
- **Stressors** can be a life or a death

Yukon Server Features

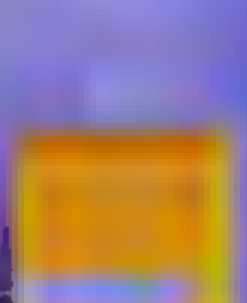
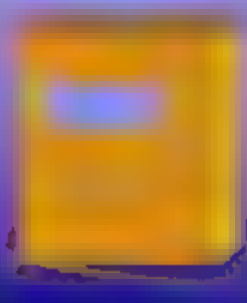
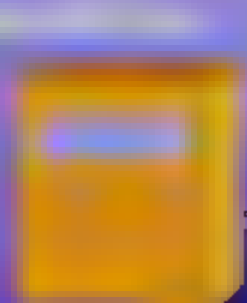
- All-in-one architecture
- Scalable architecture
- High performance
- Easy to install, manage, and maintain
- Flexible architecture

Access Methods

- **Horizontal partitioning** (sharding)
 - Each table is divided into horizontal partitions
 - Each partition is stored on a different server
 - Each partition is managed independently
- **Vertical partitioning** (columnar storage)
 - Each table is divided into vertical partitions
 - Each partition is stored on a different server
 - Each partition is managed independently
- **Indexing**
 - **B-tree** (balanced tree)
 - **Hash** (hash table)
 - **Bitmap** (bitmap index)
 - **Full-text** (full-text index)
 - **Geospatial** (geospatial index)
 - **Time-series** (time-series index)
 - **Columnar** (columnar index)
 - **Compressed** (compressed index)
 - **Unique** (unique index)

Access Methods

- **Direct Access** – Access to a specific record without having to traverse the entire file.
- **Sequential Access** – Access to a record by traversing the file from the beginning to the end.



Access Methods

Access Methods

- Access Methods: **Sequential Access Method**, **Direct Access Method**, **Indexed Access Method**
- Access Methods: **Sequential Access Method**, **Direct Access Method**, **Indexed Access Method**
- Access Methods: **Sequential Access Method**, **Direct Access Method**, **Indexed Access Method**

Prefix/Suffix Truncation

- **Prefix Truncation**
 - **Prefix Truncation** is the process of removing the prefix from a word.
 - **Prefix Truncation** is used to create new words by removing the prefix from an existing word.
 - **Prefix Truncation** is used to create new words by removing the prefix from an existing word.
 - **Prefix Truncation** is used to create new words by removing the prefix from an existing word.
 - **Prefix Truncation** is used to create new words by removing the prefix from an existing word.
 - **Prefix Truncation** is used to create new words by removing the prefix from an existing word.
- **Suffix Truncation**
 - **Suffix Truncation** is the process of removing the suffix from a word.
 - **Suffix Truncation** is used to create new words by removing the suffix from an existing word.
 - **Suffix Truncation** is used to create new words by removing the suffix from an existing word.
 - **Suffix Truncation** is used to create new words by removing the suffix from an existing word.
 - **Suffix Truncation** is used to create new words by removing the suffix from an existing word.
 - **Suffix Truncation** is used to create new words by removing the suffix from an existing word.

Prefix Unique Indexes

CREATE INDEX *idx* ON *tbl* (*col1*, *col2*, *col3*, ...)

- Unique index on the first column
- Unique index on the first two columns

Prefix/Suffix Truncation

• **Prefix/Suffix Truncation** (aka **Truncation**)

• **Truncation** (aka **Truncation**)

- **Truncation** (aka **Truncation**)

• **Truncation** (aka **Truncation**)

- **Truncation** (aka **Truncation**)

- **Truncation** (aka **Truncation**)

- **Truncation** (aka **Truncation**)

Prefix Unique Indexes

UNIQUE INDEXES: TI, CI, CI, CI, CI
UNIQUE INDEXES: CI, CI, CI, CI

- Prefix Unique Indexes
- Prefix Unique Indexes

Versioning

- **Advantages of running with multiple versions**
 - **Flexibility** (the user can choose the version of the software to use)
- **Disadvantages of running with multiple versions**
 - **Complexity** (the user has to manage multiple versions of the software)
 - **Performance** (the user has to manage multiple versions of the software)
 - **Backward compatibility** (the user has to manage multiple versions of the software)
 - **Deadlocks** (the user has to manage multiple versions of the software)

Versioning

■ **Automatic incremental build systems**

- **Make** (or **make3**) (the classic choice)

■ **Incremental C++ compiler (and C compiler) support** which performs **fast** and **correct** builds by **rebuilding** or **relinking** files as needed.

■ **Don't get into the cycles**

■ **Avoids deadlocks**

Partition Support

How to make partitions work better

- Use a partitioning scheme that is appropriate for the data and the query workload
- Use a partitioning scheme that is easy to maintain
- Use a partitioning scheme that is easy to understand
- Use a partitioning scheme that is easy to implement
- Use a partitioning scheme that is easy to test

How to make partitions work better (continued)

- Use a partitioning scheme that is easy to maintain
- Use a partitioning scheme that is easy to understand
- Use a partitioning scheme that is easy to implement
- Use a partitioning scheme that is easy to test
- Supports 10's to 100's of partitions

Backup/Restore

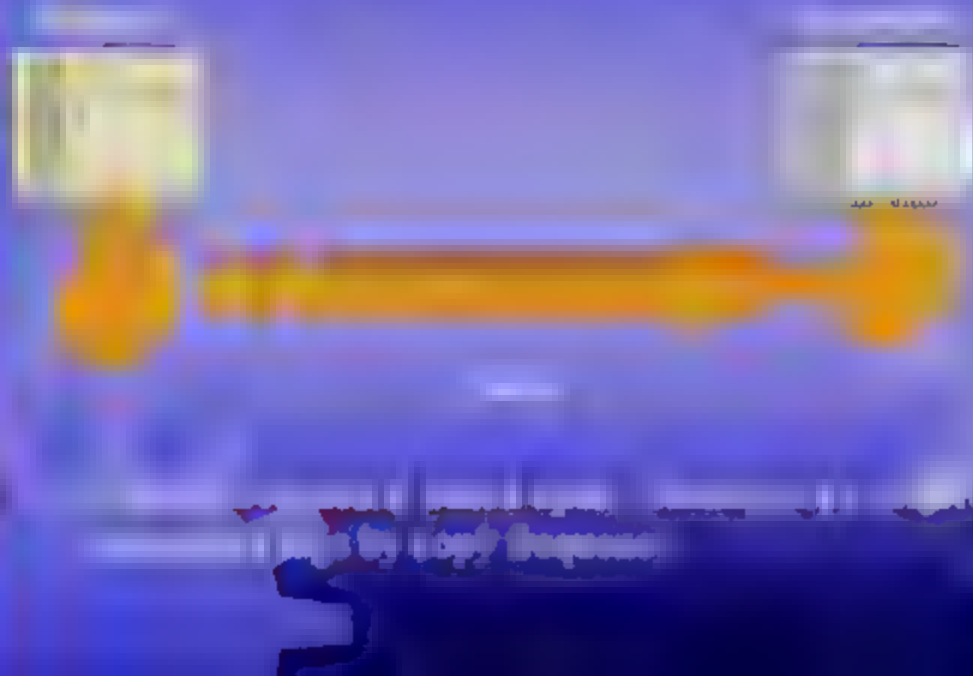
1. Backup/Restore

- Backup/Restore is a simple way to protect your data. It's a good idea to have a backup of your data in case something goes wrong.
- Backup/Restore is a simple way to protect your data. It's a good idea to have a backup of your data in case something goes wrong.
- Backup/Restore is a simple way to protect your data. It's a good idea to have a backup of your data in case something goes wrong.

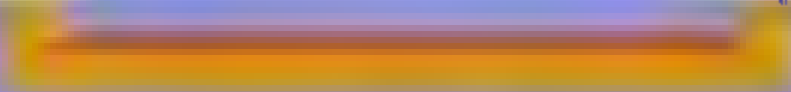
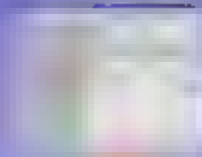
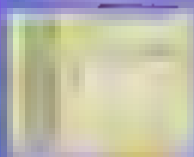
2. Backup/Restore on error

- Backup/Restore on error is a simple way to protect your data. It's a good idea to have a backup of your data in case something goes wrong.

Online log shipping



Online log shipping



Miscellaneous Availability

■ Dynamic AWE memory management

- Dynamic AWE memory management is a feature that allows the user to dynamically allocate and deallocate AWE memory.

■ Fault Tolerance

- Fault Tolerance is a feature that allows the user to configure the system to automatically restart the system in the event of a fault.
- Fault Tolerance is a feature that allows the user to configure the system to automatically restart the system in the event of a fault.

■ Remote System Administration

- Remote System Administration is a feature that allows the user to manage the system remotely.
- Remote System Administration is a feature that allows the user to manage the system remotely.

■ Dynamic AWE memory management

Relational Engine

Types of Relational Engine

- **Monolithic**
- **Modular**
- **Hybrid**
- **Cloud**
- **Embedded**
- **Standalone**
- **Open Source**
- **Proprietary**



1. **Library Services, Training, Technical Assistance**

2. **Library Services, Training, Technical Assistance**

3. **Library Services, Training, Technical Assistance**

- **Library Services, Training, Technical Assistance**
- **Library Services, Training, Technical Assistance**
- **Library Services, Training, Technical Assistance**

4. **Library Services, Training, Technical Assistance**

- **Library Services, Training, Technical Assistance**
- **Library Services, Training, Technical Assistance**
- **Library Services, Training, Technical Assistance**

5. **Library Services, Training, Technical Assistance**

6. **Library Services, Training, Technical Assistance**

XML Data Type

AD Query Support

XML Schema Support

- XML Schema is a standard for describing the structure and content of XML documents
- XML Schema is a standard for describing the structure and content of XML documents

XML Schema Support

XML Schema is a standard for describing the structure and content of XML documents

XML Schema is a standard for describing the structure and content of XML documents

XML Schema is a standard for describing the structure and content of XML documents

XML Schema is a standard for describing the structure and content of XML documents

XML Schema is a standard for describing the structure and content of XML documents

XML Schema is a standard for describing the structure and content of XML documents

Queuing Support

• **Queueing Support** | **Queueing Support** | **Queueing Support**

• **Queueing Support** | **Queueing Support** | **Queueing Support**

- **Queueing Support** | **Queueing Support** | **Queueing Support**
- **Queueing Support** | **Queueing Support** | **Queueing Support**
- **Queueing Support** | **Queueing Support** | **Queueing Support**

• **Queueing Support** | **Queueing Support** | **Queueing Support**

• **Queueing Support** | **Queueing Support** | **Queueing Support**

XML Data Type

■ AD Query Support:

■ XML as Column datatype

- XML Schema (XSD) is required
- XML Schema (XSD) is required

■ AD Query Support: XML Schema (XSD) is required

XML Schema (XSD) is required

XML Schema (XSD) is required

XML Schema (XSD) is required

XML Schema (XSD) is required

XML Schema (XSD) is required

Notification/Eventing

Online based notification areas

• User can be notified via Email, SMS, or Push

• User can be notified via Email, SMS, or Push

• User can be notified via Email, SMS, or Push

• User can be notified via Email, SMS, or Push

• User can be notified via Email, SMS, or Push

• User can be notified via Email, SMS, or Push

• User can be notified via Email, SMS, or Push

• User can be notified via Email, SMS, or Push

• User can be notified via Email, SMS, or Push

Statement Recompile

Today: (1) (2) (3) (4) (5) (6) (7) (8) (9) (10) (11) (12) (13) (14) (15) (16) (17) (18) (19) (20) (21) (22) (23) (24) (25) (26) (27) (28) (29) (30) (31) (32) (33) (34) (35) (36) (37) (38) (39) (40) (41) (42) (43) (44) (45) (46) (47) (48) (49) (50) (51) (52) (53) (54) (55) (56) (57) (58) (59) (60) (61) (62) (63) (64) (65) (66) (67) (68) (69) (70) (71) (72) (73) (74) (75) (76) (77) (78) (79) (80) (81) (82) (83) (84) (85) (86) (87) (88) (89) (90) (91) (92) (93) (94) (95) (96) (97) (98) (99) (100)

- Statement recompile
- Statement recompile
- Statement recompile
- Statement recompile

Statement recompile on unexecuted branch

Fine Grained Security

- **Open Group, International Standard Security**
 - **ISO 15400** (Security Framework)
 - **ISO 15401** (Information Security Management System)
 - **ISO 15402** (Information Security Management System)

■ **Information Security**

Information Security is the protection of information from unauthorized access, use, disclosure, modification, or destruction.

Information Security is the protection of information from unauthorized access, use, disclosure, modification, or destruction.

Information Security is the protection of information from unauthorized access, use, disclosure, modification, or destruction.

Information Security is the protection of information from unauthorized access, use, disclosure, modification, or destruction.

Information Security is the protection of information from unauthorized access, use, disclosure, modification, or destruction.

T-SQL Exception Handling

■ **TRY...CATCH** (introduced in SQL Server 2005)

■ **TRY...CATCH** is used to handle errors in T-SQL

■ **TRY...CATCH** is used to handle errors in T-SQL

■ **TRY...CATCH** is used to handle errors in T-SQL

■ **TRY...CATCH** is used to handle errors in T-SQL

■ **TRY...CATCH** is used to handle errors in T-SQL

■ **TRY...CATCH** is used to handle errors in T-SQL

■ **TRY...CATCH** is used to handle errors in T-SQL

■ **TRY...CATCH** is used to handle errors in T-SQL

Recursive Query Example

▶ **UNIT 14: RECURSIVE QUERY**

QUESTION

Write a recursive query to find the names of all employees who have worked on projects that have been completed. The query should return the employee's name, the project name, and the completion date. The query should be able to handle multiple levels of hierarchy in the project structure.

ANSWER

The following SQL query uses a recursive common table expression (CTE) to find all employees who have worked on completed projects. The query starts with a base case that identifies employees who have worked on projects that are marked as completed. It then recursively joins the table to itself to find all other employees who have worked on the same project.

Full Text Indexing: Yukon Features

- Performance and Scalability
 - Re-architected
- Integration with SQL Server
 - Full DDL support
 - Backup, restore, & recovery
 - Fulltext queries against views & linked servers
 - Attach/detach, removable DB
 - Replication of FT changes

Full Text Indexing: Yukon Features

- Performance and Scalability
 - Re-architected
- Integration with SQL Server
 - Full DDL support
 - Backup, restore, & recovery
 - Fulltext queries against views & linked servers
 - Attach/detach, removable DB
 - Replication of FT changes



Follow up

- Before we knew what we were going to build and needed your help to make sure we built it right.
- Now, we need your help in building the right thing.
- We'll make a winner - together.



Follow up

- Before we knew what we were going to build and needed your help to make sure we built it right.
 - Now, we need your help in building the right thing.
 - We'll make a winner - together.
- 